Positionnement – les techniques actuelles

Introduction

De nombreuses techniques de positionnement existent, parfois anciennes et à proscrire, d'autres moins connues, d'autres encore récentes et « très en vogue ».

La propriété CSS `display` propose de nombreuses possibilités, plus de 40 valeurs possibles, chaque valeur définissant le type de boite utilisée pour le rendu de l'élément, du simple inline, …, au modèle de boite flexible en passant par le modèle tabulaire ou le modèle de grille.

La nécessité de disposer d'une technique simple à utiliser, **prévisible sur différentes tailles d'appareils** (site web adaptatif ou responsive web design) a amené à la définition en CSS3 du modèle à base de boite flexible (`Flexbox`) aujourd'hui très en vogue que l'on peut directement mettre en œuvre.

Une alternative consiste à utiliser des **classes CSS prédéfinies par des communautés**, fournies dans des fichiers CSS (avec souvent du JavaScript), que l'on peut, sous réserve des licences prévues, utilisées.

On trouve ainsi aujourd'hui de nombreux « frameworks CSS » tels que :

- **Bootstrap** : « The most popular HTML, CCS and JavaScript framework for developing responsive, mobile first projects on the web";
- **Foundation** : "The most advanced responsive front-end framework in the world";
- **Bulma** : "An open source CSS framework based on Flexbox";
- **Ulkit** : "A lightweight and modular front-end framework for developing fast and powerful web interfaces";
- **Semantic UI** : "A UI component framework based around useful principles from natural language"; Ou encore Clank, Skeleton, KickStart, Kube, PureCSS, KNACSS, GroundworkCSS, Cascade, Jeet, Milligram, … Cf : https://www.codeur.com/blog/front-end-framework/) • …

Mais avant tout, il faut comprendre comment CSS3 permet la création de site web adaptatif.

Les Media Queries

https://www.alsacreations.com/article/lire/930-css3-media-queries.html

La spécification CSS3 `Media Queries` définit les techniques pour l'application de feuilles de styles en fonction des périphériques de consultation utilisés pour du HTML.

Ces bonnes pratiques permettent d'exploiter encore plus les avantages de la séparation du contenu et de la présentation : **l'intérêt est de pouvoir satisfaire des contraintes de dimensions, de résolutions et d'autres critères variés pour améliorer l'apparence graphique et la lisibilité (voire l'utilisabilité) d'un site web**.

Toutes les plateformes sont concernées : navigateurs mobiles et tablettes, écrans à faibles résolutions, impression, tv, synthèses vocales, plages braille, etc. L'attribut `media` de la balise `link` ou la directive `@media` d'une feuille de style CSS permettent de préciser le contexte d'application des règles de style ;

les valeurs possibles sont : screen (écran), handheld (périphérique mobile de petite taille), print (impression), aural ou speech (synthèse vocale), braille (plage braille), embossed (imprimante braille), projection (projecteur), tv

(téléviseur), tty (terminal).

Pour un contexte, il est possible d'adjoindre un panel ce critères plus précis à l'aide de propriétés et de valeurs numériques, de préfixes `min-` et `max-` pour les critères numériques ainsi que de combinaisons multiples de ces mêmes critères. Le but est de cibler plus finement les périphériques de destination en fonction de leurs capacités intrinsèques. Parmi les critères, on peut citer : orientation (orientation du périphérique : portait ou landscape), résolution (résolution du périphérique en dpi, dppx, ou dpcm), min-width, width ou max-width (dimension en largeur de la zone d'affichage), height (dimension en hauteur de la zone d'affichage), device-width (dimension en largeur du périphérique), …

Question 1

1. Créez une page html comportant une balise nav suivi d'un paragraphe. Placez dans la partie navigation une balise ul comportant 5 items.
2. A l'aide de la directive `@media`, faites que les items s'affichent horizontalement pour les largeurs d'affichages d'au moins 768px, verticalement pour les zones d'affichage plus petites, ne s'affichent pas lors de l'impression.

Le modèle de boite flexible

On donne

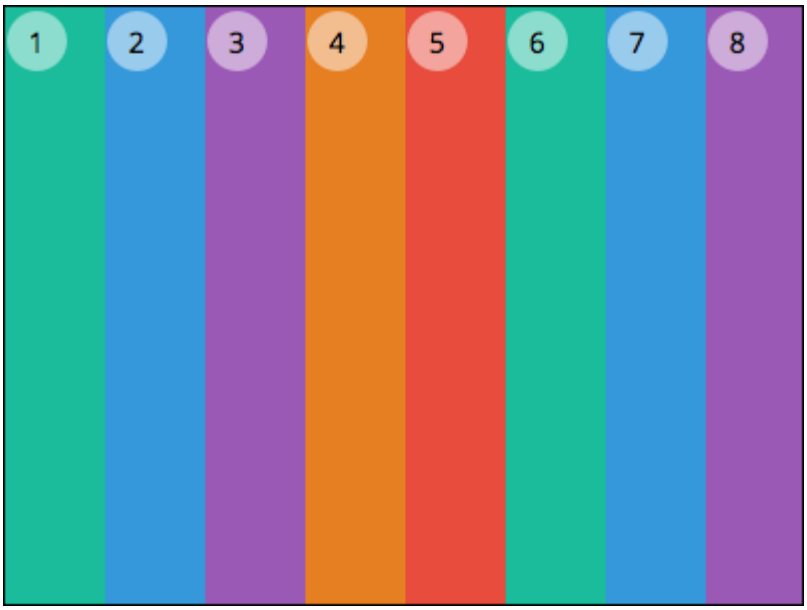
```
 <section>
        <div><span>1</span></div>
        <div><span>2</span></div>
        <div><span>3</span></div>
        <div><span>4</span></div>
        <div><span>5</span></div>
        <div><span>6</span></div>
        <div><span>7</span></div>
        <div><span>8</span></div>
        <div><span>9</span></div>
</section>
```

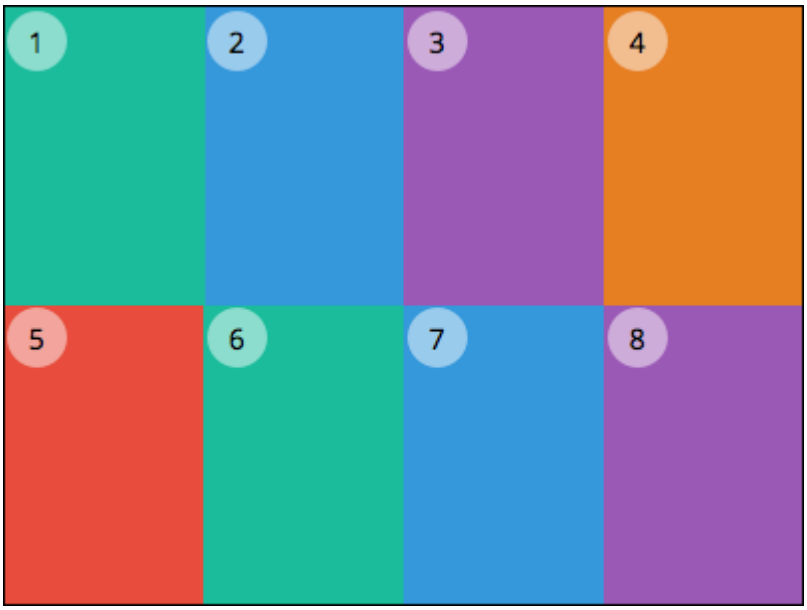
et le css de base :

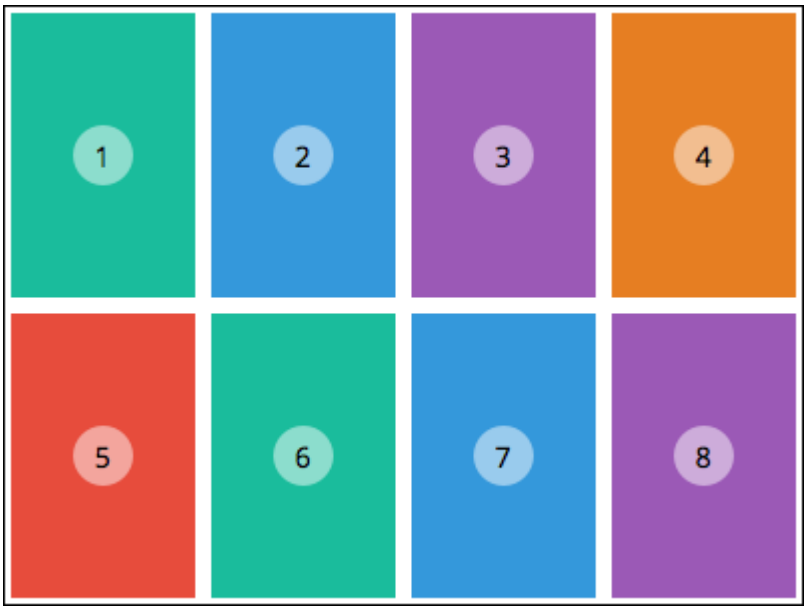
```
/* ====================================
   CSS setup ne rien modifier
   Editer plus bas
 * ==================================== */

body { margin: 0; padding: 0; border: 0; }
section { min-height: 100vh; }
div { background-color: lightgrey; }
div:nth-of-type(5n+1) { background-color: #1abc9c; }
div:nth-of-type(5n+2) { background-color: #3498db; }
div:nth-of-type(5n+3) { background-color: #9b59b6; }
```

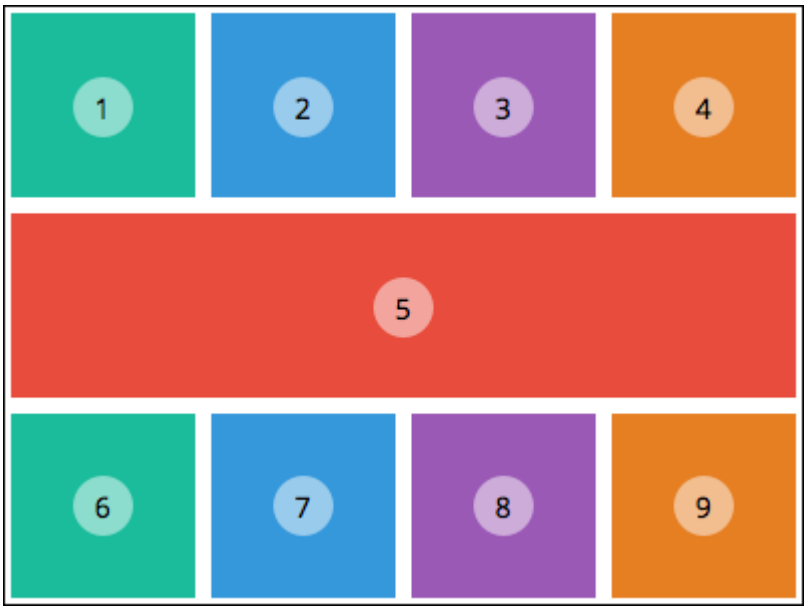
```
div:nth-of-type(5n+4) { background-color: #e67e22; }
div:nth-of-type(5n) { background-color: #e74c3c; }
span { background-color: rgba(255, 255, 255, .5); border-radius: 50%;
display: block; font-size: 48px; height: auto; line-height: 3em; text-
align: center; width: 3em; }

/* ====================================
   Editer a partir d'ici
 * ==================================== */
```

Obtenir successivement :

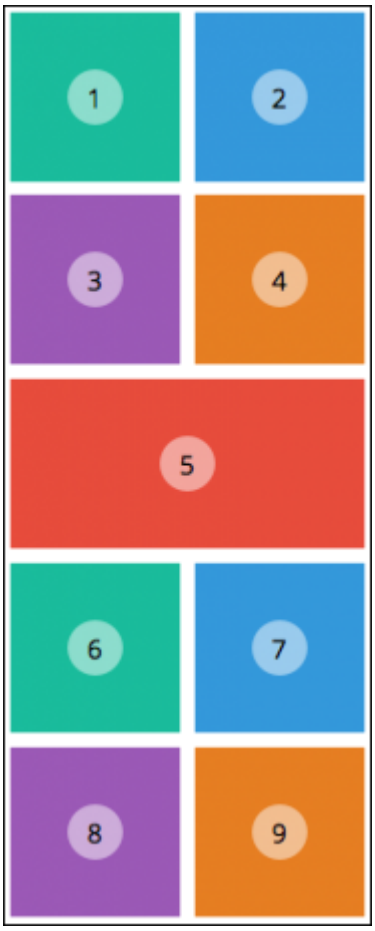




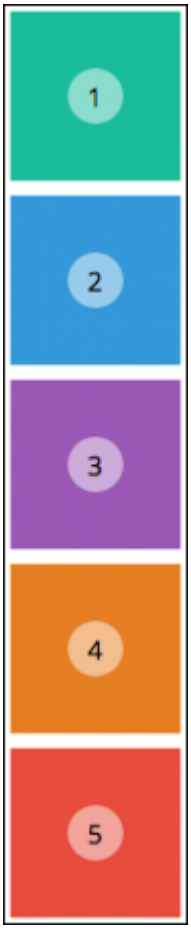




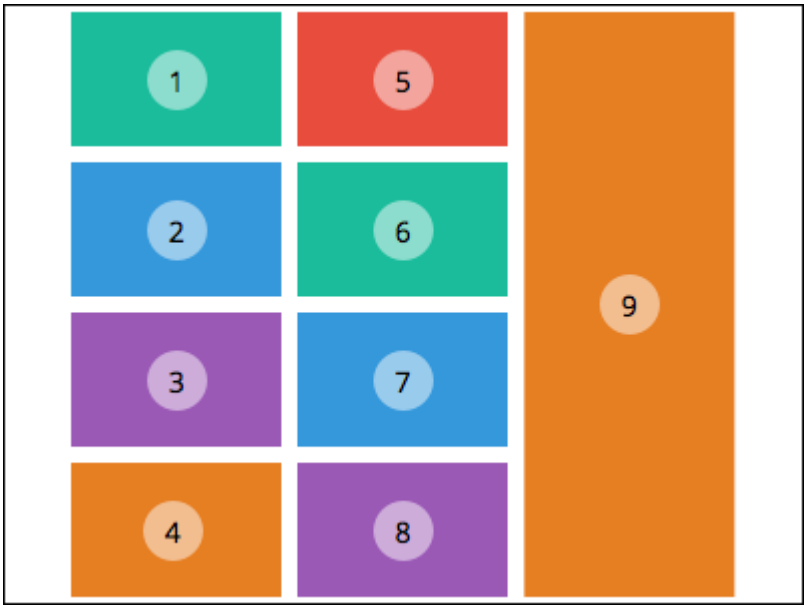
modifier la composition précédante avec un breakpoint à 1024px de large pour obtenir (en dessous de 1024 px)



Modifier la composition précédante avec un breakpoint à 600px de large.



Répartir les contenus sur 2 à 3 colonnes centrées (limiter la largeur de la `<section>` parente au besoin).
Changer la hauteur du viewport et observer le comportement d'affichage.



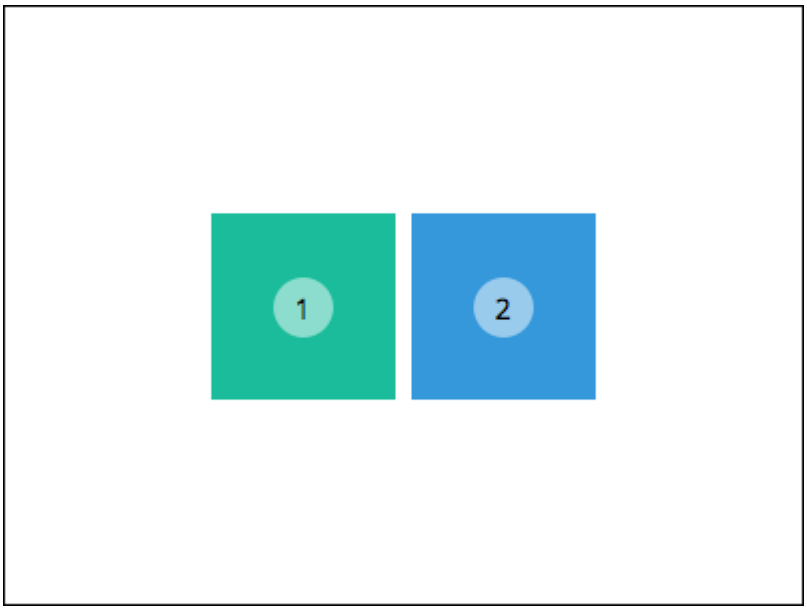
Maintenant, avec la base html

```
<section>
        <div><span>1</span></div>
        <div><span>2</span></div>
</section>
```

et css

```
body { margin: 0; padding: 0; border: 0; }
section { min-height: 100vh; }
div { background-color: lightgrey; }
div:nth-of-type(5n+1) { background-color: #1abc9c; }
div:nth-of-type(5n+2) { background-color: #3498db; }
div:nth-of-type(5n+3) { background-color: #9b59b6; }
div:nth-of-type(5n+4) { background-color: #e67e22; }
div:nth-of-type(5n) { background-color: #e74c3c; }
span { background-color: rgba(255, 255, 255, .5); border-radius: 50%;
display: block; font-size: 48px; height: auto; line-height: 3em; text-
align: center; width: 3em; }
```

Obtenir :



Avec la base :

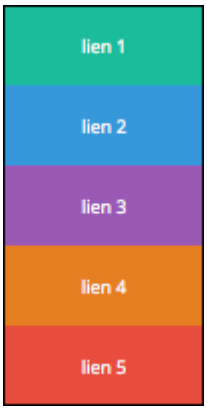
```
<ul>
        <li><a href="#">lien 1</a></li>
        <li><a href="#">lien 2</a></li>
        <li><a href="#">lien 3</a></li>
        <li><a href="#">lien 4</a></li>
        <li><a href="#">lien 5</a></li>
</ul>
```

et css

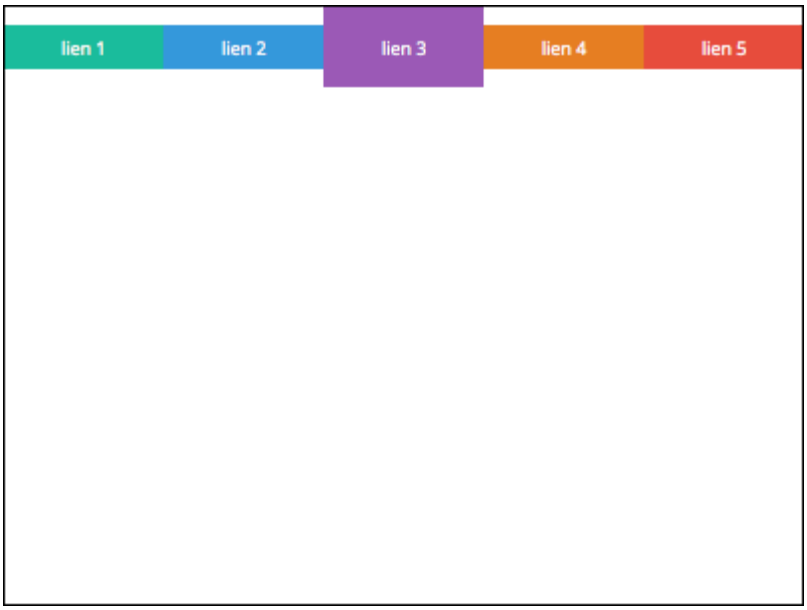
```
body { margin: 0; padding: 0; border: 0; }
ul { list-style: none; padding: 0; margin: 0; height: 100px; }
li { background-color: lightgrey; text-align: center; }
li:nth-of-type(5n+1) { background-color: #1abc9c; }
li:nth-of-type(5n+2) { background-color: #3498db; }
li:nth-of-type(5n+3) { background-color: #9b59b6; }
li:nth-of-type(5n+4) { background-color: #e67e22; }
li:nth-of-type(5n) { background-color: #e74c3c; }
a { color: white; font-size: 16px; line-height: 50px; text-decoration:
none; }
```

Obtenir le menu :

Présenter le menu à la verticale avec un breakpoint à 600px.



au survol, étirer le `<li>`.



Et pour finir, avec la base :

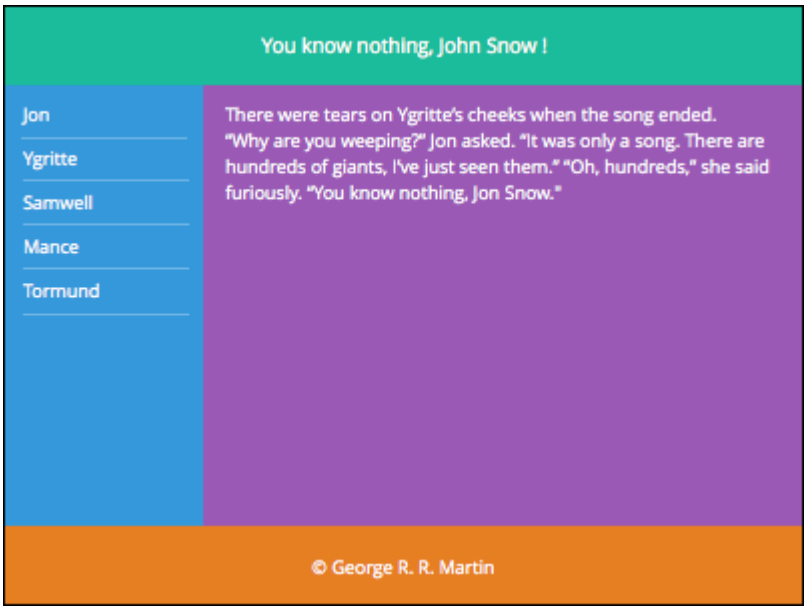
```
<header>
        <h1>You know nothing, John Snow !</h1>
</header>

<main>
        <nav role="navigation">
                <a href="#">Jon</a>
                <a href="#">Ygritte</a>
                <a href="#">Samwell</a>
                <a href="#">Mance</a>
                <a href="#">Tormund</a>
        </nav>
        <article>
                <p> <p>There were tears on Ygritte’s cheeks when the song
ended. <q>Why are you weeping ?</q> Jon asked. <q>It was only a song.
There are hundreds of giants, I’ve just seen them.</q> <q>Oh, hundreds,
</q> she said furiously. <q>You know nothing, Jon Snow…</q></p></p>
        </article>
</main>

<footer>
        <p> George R. R. Martin</p>
</footer>
```

et le css:

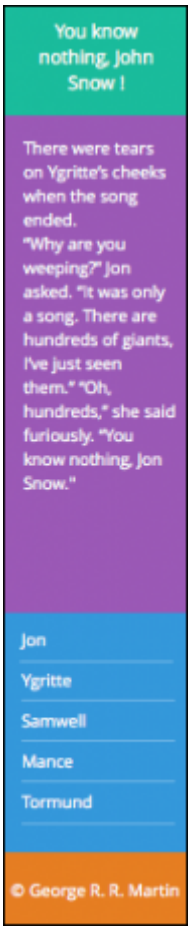
```
body { border: 0; margin: 0; padding: 0; }
header { background-color: #1abc9c; line-height: 80px; }
nav { background-color: #3498db; }
article { background-color: #9b59b6; }
footer { background-color: #e67e22; }
header, nav, article, footer { color: white; padding: 10px; }
header, footer { height: 80px; text-align: center; }
h1, p { margin: 0 0 15px 0; }
nav a { border-bottom: 1px solid rgba(255,255,255,.3); color: white;
display: block; line-height: 40px; text-decoration: none; }
```

Les `<header>` et `<footer>` ont une hauteur fixe, les éléments contenus dans `<main>` doivent s'adapter à l'espace vertical libre.

Présenter le menu à l'horizontal en résolution supérieure à 600px. Le menu a une largeur fixe de 200px.



Les contenus se présentent à la verticale avec un breakpoint à 600px de large. Le menu passe sous l'article.



Framework CSS, un exemple Bootstrap

> Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery https://getbootstrap.com

La version utilisée dans la suite est Bootstrap en version 4.1.1 (Avril 2018). Attention, la version 4 comporte des changements importants rendant inopérants de nombreuses classes des versions précédentes. Assurez-vous, que vous consultez bien une documentation correspondant à cette version.

Il existe de très nombreux tutoriaux sur Bootstrap ; on peut citer par exemple :

- https://www.w3schools.com/bootstrap4/ : tutoriel Bootstrap 4 avec de nombreux exemples
- ou en vidéo : https://www.youtube.com/watch?v=gm2RCfjXS3s « Bootstrap 4 Get Started »

Quelques éléments pour débuter :

> If you don't want download and host Bootstrap 4 yourself, you can include it from a CDN (Content Delivery Network):

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/css/bootstrap.min.cs
s">
```

> To ensure proper rendering and touch zooming, add the following `<meta>` tag inside the `<head>` element:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

> Bootstrap 4 also requires a containing element to wrap site contents. There are two container classes to choose from:
>
> - The .container class provides a responsive fixed width container
> - The .container-fluid class provides a full width container, spanning the entire width of the viewport.

Question

- Créez une page simple utilisant Bootstrap 4 en le téléchargeant ou en utilisant un réseau de diffusion de contenu (« CDN : Content Delivery Network ») illustrant la différence entre conteneur fixe et conteneur fluide.

  > Vous devez constater que la classe container centre le bloc sur une largeur inférieur à 510 puis fixe à 510, puis à 690, puis à 930, puis à 1110, ... , qui s'adapte donc en fonction de la largeur de l'écran. La classe container-fluid permet au bloc d'occuper en permanence toute la largeur disponible.

- « Bootstrap 4 Grid System » Bootstrap repose sur un **système de grille permettant de découper un élément de classe row en 12 colonnes**, colonnes qui peuvent être utilisées seules ou groupées. Pour s'adapter aux différentes tailles d'écran, Bootstrap utilise 5 préfixes de classes :
  - col- (extra small devices - screen width less than 576px)
  - col-sm- (small devices - screen width equal to or greater than 576px)
  - col-md- (medium devices - screen width equal to or greater than 768px)
  - col-lg- (large devices - screen width equal to or greater than 992px)
  - col-xl- (xlarge devices - screen width equal to or greater than 1200px)

Suffixé par une valeur entre 1 et 12, cela permet de préciser la taille occupée par un élément dans la ligne selon une taille d'écran. Par exemple, appliquer `class="col-sm-3 col-lg-2"` permettra à l'élément d'occuper un quart de l'espace de son parent de classe row pour les écrans de taille au moins égale à 576px mais seulement un sixième de l'espace à partir des écrans de taille au moins égale à 992px.

Exemple de code :

```
<div class="container">
        <div class="row">
                <div class="col-sm-1 col-md-3">This is part of our grid.
</div>
                <div class="col-sm-5 col-md-3">This is part of our grid.
</div>
                <div class="col-sm-3 col-md-3">This is part of our grid.
</div>
                <div class="col-sm-3 col-md-3">This is part of our grid.
</div>
</div >
        <div class="row">
                <div class="col-2 col-lg-1">This is part of our grid.
</div>
                <div class="col-2 col-lg-10">This is part of our grid.
</div>
                <div class="col-4 col-lg-1">This is part of our grid.
</div>
        </div >
</div>
```

Question

- Recopiez le code précédent, faites varier la taille de la fenêtre. Appliquez une bordure rouge aux blocs div à l'intérieur d'un élément de classe row afin d'observer l'espace occupé. Jouez un peu avec les préfixes et les tailles.
- Que se passe-t-il quand la somme est inférieure à 12 ? supérieure à 12 ? Il est bien entendu possible de définir des row à l'intérieur d'une autre row.

Question

Placez un bloc de classe row dans le second bloc de la seconde ligne. A l'intérieur de ce bloc, placez 12 éléments de taille 1 pour toute taille d'écran. Il est possible de dire, selon les différentes configurations, les blocs qui doivent ou non s'afficher. Attention, Bootstarp 4 ne définit plus les classes préfixées par hidden ou visible des versions précédentes. Dans sa dernière version, le prefixe utilisé est « d » pour display ; une balise dont la classe est d-md-none ne sera pas affichée si la taille de la fenêtre correspond à md. Une balise dont la classe est d-md-block (mode block) (ou d-md-inline ou d-md-…) sera affichée si la taille de la fenêtre correspond à md. Il est également assez simple de définir les zones imprimables des zones qui ne le sont pas (d-print-none ou d-print-block par exemple).

Question

Recopiez et complétez le code ci-dessous afin que les blocs « Visible only ... » n'apparaissent que lorsque la taille de la fenêtre du navigateur correspond.

```
<div class ="container" style="padding:40px;">
        <div class="row">
                <div class ="col-4 col-sm-2" style="background-color:
yellow;
                box-shadow:inset 5px 5px 5px white, inset -5px -5px 5px
white;">
                        <span class="d-none d-sm-block">Hidden</span>
                        <span class="d-block d-sm-none">Visible only
Extra-small </span>
                </div>
                <div class="col-2 col-sm-4 col-md-2" style="background-
color: yellow;
                                box-shadow:inset 5px 5px 5px white, inset
-5px -5px 5px  white;">
                        <span class="d-block d-sm-none d-md-
block">Hidden</span>
                <span class="                            ">Visible only
Small </span>
        </div>
        <div class="col-2 col-md-4 col-lg-2" style="background-color:
yellow;
        box-shadow: inset 5px 5px 5px white, inset -5px -5px 5px  white;">
                <span class="                            ">Hidden</span>
                <span class="                            ">Visible only
Medium </span>
        </div>
        <div class="col-2 col-lg-4 col-xl-2" style ="background-color:
yellow;
        box-shadow : inset 5px 5px 5px white, inset -5px -5px 5px
white;">
                <span class="                             ">Hidden</span>
                <span class="                             ">Visible only
Large </span>
        </div>
        <div class="col-2 col-xl-4" style ="background-color: yellow;
        box-shadow : inset 5px 5px 5px white, inset -5px -5px 5px
white;">
                <span class="                           ">Hidden</span>
                <span class="d-none d-xl-block">Visible only Extra large
</span>
        </div>
</div>
```

Il est possible de réordonner les éléments en utilisant le préfixe `order`. Par exemple `order-sm-1` placera cet élément en première position si la taille de la fenêtre correspond à sm.

Question

Créez un bloc incluant cinq blocs contenant chacun une des valeurs entre 1 et 5, affichés en ordre croissant dans le cas d'une fenêtre extra small, dans l'ordre décroissant pour les autres configurations.

Pour toute question concernant la définition des marges, l'alignement vertical, … n'hésitez pas à consulter la documentation, notamment le chapitre « Utilities » : https://getbootstrap.com/docs/4.0/utilities/borders/